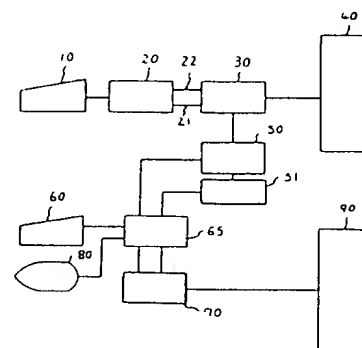


(54) SEQUENCE CONTROL PROGRAM GENERATING DEVICE

(11) 62-202202 (A) (43) 5.9.1987 (19) JP
 (21) Appl. No. 61-43988 (22) 3.3.1986
 (71) HITACHI LTD(1) (72) MASAOKI TAKAGI(1)
 (51) Int. Cl. G05B19/02

PURPOSE: To improve the programming work efficiency by editing, correcting, and storing reusable programs in an area other than a program area of already developed programs.

CONSTITUTION: Reusable programs out of already developed programs stored in an already developed program storage memory 40 are designated as an output file by a read circuit designating input device 10. The name of this output file is converted automatically to a machine instruction words by an instruction word converter 20 and is transmitted to a coincident instruction word detector 30. The detector 30 compares and collates successively programs stored in the memory 40. When the designated instruction word is found, an objective sequence is cut out by a sequence block cutting-out device 50 and is stored in a sequence block buffer memory 51. Simultaneously, it is displayed on a display device 80 and is confirmed by an operator. When the confirming operation of reusable programs is terminated, they are registered as reusable programs in a reusable program storage memory 90 by an input device 60.



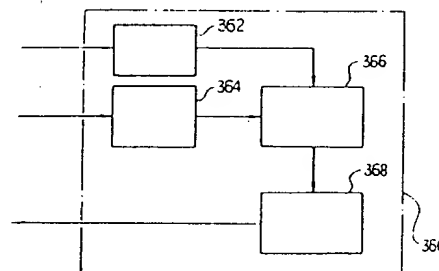
21: detection start signal, 65: circuit diagram converter.
 70: sequence block register device

(54) METHOD FOR DETERMINING ORIGIN OF WORKING OF ARTICULATED MANIPULATOR

(11) 62-202203 (A) (43) 5.9.1987 (19) JP
 (21) Appl. No. 61-43715 (22) 28.2.1986
 (71) HITACHI LTD (72) MAKOTO SENOO(5)
 (51) Int. Cl. G05B19/18, B25J9/16, B25J13/00

PURPOSE: To determine the position of the best working origin by calculating a set of positions of working origins of an articulated manipulator and selecting the position of the working origin, which maximizes or minimizes the value of an evaluation function, from this set.

CONSTITUTION: In a working origin position determining device 360, data inputted from a console before the start of operation is stored in an obstacle three-dimensional shape storage part 362. New obstacles, which are found when the hand of the articulated manipulator is guided to working points, are stored as data also. A coordinate point P_i of the i -th working point is inputted to and stored in a hand operation pattern storage part 364. An evaluation function arithmetic part 366 calculates the position of the best working origin of the manipulator which maximizes the evaluation function for the working point P_i and a hand working area corresponding to the point P_i . A best working origin position operating part 368 searches and determines the working origin, to which the operated evaluation function gives a maximum value, out of working origins. Thus, the position of the best working origin for working points is determined.

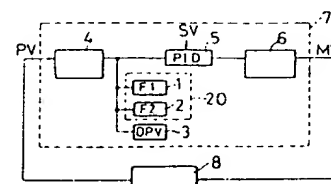


(54) PROPORTIONAL INTEGRAL AND DIFFERENTIAL CONTROLLER WITH PROCESS UPPER AND LOWER LIMIT ALARM DEVICE

(11) 62-202204 (A) (43) 5.9.1987 (19) JP
 (21) Appl. No. 60-216355 (22) 30.9.1985
 (71) MITSUBISHI ELECTRIC CORP (72) TOSHIAKI NAKAHARA
 (51) Int. Cl. G05B23/02, G05B11/42

PURPOSE: To set an optimum range in accordance with the operation of a process by giving preliminarily a conditional formula to process upper and lower limit alarm device and comparing this conditional formula with a feedback signal value.

CONSTITUTION: A feedback signal PV from a process 8 is converted to a numerical value for operation by an input processing circuit 4 and is inputted to a PID (proportional integral and differential) arithmetic processing circuit 5. The circuit 5 operates a manipulated variable MV so that a deviation DV between a given target value SV and the signal PV is reduced. A process upper and lower limit alarm processing circuit 20 outputs an alarm if the inputted signal PV is larger than the upper limit value or smaller than the lower limit value. The circuit consists of a conditional formula processing circuit 1 for process upper limit alarm and a conditional formula processing circuit 2 for lower limit alarm, and it is possible to set conditional formulas F1 and F2 consisting of one of prescribed conditional formulas or combination of plural conditional formulas to circuits 1 and 2. Thus, an optimum alarm range is provided in not only the fixed command control but also the variable value control and the batch limitation.



6: output processing circuit

⑫ 公開特許公報(A)

昭62-202202

⑩ Int.Cl.⁴

識別記号

庁内整理番号

⑬ 公開 昭和62年(1987)9月5日

G 05 B 19/02

P-7740-5H

審査請求 未請求 発明の数 1 (全15頁)

⑭ 発明の名称 シーケンス制御プログラム作成装置

⑮ 特 願 昭61-43988

⑯ 出 願 昭61(1986)3月3日

⑰ 発 明 者 高 木 正 興 日立市大みか町5丁目2番1号 株式会社日立コントロールシステムズ内

⑱ 発 明 者 野 中 広 人 日立市大みか町5丁目2番1号 株式会社日立コントロールシステムズ内

⑲ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑳ 出 願 人 日立コントロールシステムズ 日立市大みか町5丁目2番1号

㉑ 代 理 人 弁理士 小川 勝男 外2名

明 題 書

1. 発明の名称

シーケンス制御プログラム作成装置

2. 特許請求の範囲

1. シーケンス制御を行なうプログラマブルコントローラのシーケンス制御プログラム作成装置において、既作成プログラム格納メモリから標準制御回路として再利用できる回路ブロックを切出して格納する再利用プログラム格納メモリを備え、該回路ブロックを修正・編集してシーケンス制御プログラムを作成するシーケンス制御プログラム作成装置。

2. 特許請求の範囲第1項において、前記回路ブロックを切出す際に、回路ブロックを指定する入力装置、回路ブロックを既作成プログラム格納メモリより検出する一致命令検出装置、回路ブロックを切出すシーケンスブロック切出し装置を備え、該装置を用いて回路ブロックを切出すことを特徴とするシーケンス制御プログラム作成装置。

(1)

3. 特許請求の範囲第1項において、前記回路ブロックのうち、制御要素のナンバー、名称全体又はリレーシンボルの修正を行なうことを特徴とするシーケンス制御プログラム作成装置。

4. 特許請求の範囲第1項において、前記回路ブロックを修正した後再利用プログラム格納メモリに登録するためのシーケンスブロック登録装置を備えたことを特徴とするシーケンス制御プログラム作成装置。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明はプログラマブルコントローラ（以下P.C.と略す）に係り、特にプログラマブルコントローラのシーケンス制御プログラム作成装置に関する。

〔従来の技術〕

従来、P.C.のプログラム作成においては「よくわかるプログラマブルコントローラ」（電気学院出版）の第3章にあるように、リレー回路図イメージで対話方式により作成していくため、プロ

(2)

プログラムの作成自体は、非常に容易でわかりやすいものであつた。しかし、作成したプログラムを、その後、何度も再利用することについては、何ら配慮がなされていなかった。

尚、この種の装置に関するものとしては特開昭60-63603号などがある。

〔発明が解決しようとする問題点〕

シーケンス制御は、機器制御の中でも簡単であるため、非常によく使用される。特に制御装置が従来のリレー盤より、電子式のプログラマブルコントローラが安価に市場に供給されるようになって以来、PCsの普及、浸透にはめざましいものがある。

制御ロジックが、リレー回路からPCsのソフトに変った結果、プログラムの作成、デバッグ、

正が極めて容易となつた。しかしながら、高級PCsは、本来マイクロコンピュータの機能をもあわせもっており、このマイクロコンピュータの機能を活用することによって、徹底的にプログラムの作成効率を向上することを目指したものが本

(3)

データベースの標準化では、再利用できても、再びプログラム、及び、図面の作成作業は、行なわなければならない、設計効率向上には限界があつた。

本発明の目的は、既開発終了したプログラムの中から、再利用可能なプログラムを抽出し、これを既開発のプログラムエリアとは、全く異なつたメモリエリアに、再利用可能なプログラムとして登録格納することにより、従来の書類ベースのプログラム標準化をプログラムそのものの形体で行うことにある。その結果、標準プログラムの作成作業、すなわち、プログラム作成装置を使つて、標準図面を参照しながら人手で打込むという作業を排除し、設計生産効率を大幅に向上させようというものである。

〔問題点を解決するための手段〕

書類ベースのプログラム標準化では、前述の通り、設計効率向上に限界があるため、本発明はプログラムそのものを、標準化することで、設計効率を向上させるものであり、そのためには、既開発終了したプログラムの中から、再利用可能

(5)

発明である。

PCsの制御プログラムは、通常高級機であれば、リレー回路図イメージで、マンマシンコミュニケーションを行ないながら作成していく。ところで、この制御プログラムは、制御対象が類似しているものに対しては、当然のことながら、その制御プログラムも類似したものとなる。大型の工作機械であるトランスファマシンの例に上げると、工作機械自体は、ユーザーの要求によって仕様が異なるため、当然、シーケンスプログラム自体も様々になるわけであるが、一般的には、60%の制御ロジックが、ロジックとして、リピート使用可能である。また一部修正することによって再利用可能なものが約20%、新規に作成しなければならないものが約20%という具合である。

そこで、再利用可能なものを活用するため、従来は、再利用可能なシーケンス回路パターンを抽出し、これを標準図面として、書類ベースで管理し、必要な時にこれを探し出し、入出力信号のつけかえなどを行つて再利用してきた。ところが、

(4)

な部分を抽出し、これを、既開発のプログラムエリアとは、全く別のエリアに格納することで達成される。

〔作用〕

既開発のプログラムエリアとは、全く別のエリアに、再利用可能なプログラムを編集・修正・格納することにより、再利用可能なプログラムが従来のような書類ベースでなく、プログラムそのもので、再利用可能となる。そのため従来の書類を参照しながらのプログラム作成の手間がはぶかれ、設計生産効率を大幅に向上できる。

〔実施例〕

本発明の概要を、具体的実施例に基づき、第1図により説明する。

第1図において、既開発プログラムは、メモリ40に格納されている。この既開発プログラムの中で再利用可能なものを、読出し回路指定入力装置10から入力することにより指定する。この装置は、一般的にはパソコンのキーボード等である。

この入力装置10により、再利用可能なものを

(6)

読出するための指定は、一般的に、指定した回路ブロックで使用されている出力コイルの名称で行なう。出力コイル名称で読出すのは、出力コイルが全回路において、唯一1回だけしか使用されないためである。この出力コイル名称は、命令語変換装置20により、PCの機械命令語に自動変換される。

この機械命令語は、命令語変換装置20より、一致命令語検出装置30に伝達されるとともに、検出開始信号21が出される。

この検出開始信号21が、一致命令語検出装置30に伝えられると、一致命令語検出装置30は既開発プログラム格納メモリ40に格納されているプログラムを、メモリ先頭アドレスより、逐次1ワードづつ、読出し指定された出力コイルの機械命令であるかどうか、比較照合していく。そして、読出し指定された命令語が見つかったと、シーケンスブロック切出し装置50に起動がかかり、その読出し指定された出力コイルが、かかわっているリレーラダー回路の先頭アドレス、および、

(7)

装置は、同一のものであつても、何ら問題はない。

さて、プログラムの登録は、今迄パツファ51に格納されていた、再利用可能な1シーケンスブロックのプログラムが、既開発プログラム格納メモリエリア40とは、全く違った、再利用プログラム格納メモリエリア90に、転送されることで成される。この、再利用プログラム格納メモリエリア90に収納する際には、当然、再利用プログラムとして多数のシーケンスブロックが既に登録されている可能性がある。そのため、登録の際は、後で述べるように、既に登録済のプログラムと、協調をとり、プログラムの再割付自動編集を行なう必要がある。

ちなみに、シーケンスブロックとは、第2図で示すA、及びBのように一つのラダー回路が一つの回路として完結しているかたまりを言う。

ここで、一致命令検出装置30の処理内容を、第3図で説明する。

命令語変換装置20より、一致命令検出装置30に対して、検索しようとする命令語が、検索

(9)

最終アドレスが自動検出され、シーケンスブロック切出し装置50により、目的のシーケンスブロックが自動的に切出され、シーケンスブロックバッファメモリ51に格納される。同時に、このシーケンスブロックのプログラムを、回路図変換装置65にかけ、機械命令語から回路図に変換後、回路図として、表示装置80に表示し、オペレータの確認を行なう。

もしこの時、オペレータが要求した回路でなかった場合は、再度、要求回路の読出し要求操作を行なえばよい。また、読出された回路の一部に変更、修正をかけるには、この時、パツファ51上に読出されている機械語に対して、プログラム修正の操作を行なえばよい。

このように、再利用可能なプログラムであることの確認操作が終了すると、次に、再利用可能なプログラムとして登録する。この登録は、入力装置60から、起動信号を入力することで、シーケンスブロック登録装置70に起動がかかり行なわれる。尚、読出し回路指定入力装置10と、入力

(8)

命令語レジスタ301にセットされる。そして、一致比較装置302に対して、検出開始信号21がはいる。すると命令語読出し制御装置303に起動がかかり、アプリケーションシーケンスプログラムが格納されているプログラム先頭アドレスを記憶している先頭アドレスレジスタ304の内容が、読出しポインタ305にセットされる。そして、命令語読出し制御装置303によつて、既開発プログラムメモリ40より読出しポインタ305が示しているアドレスの内容を取出し、読出し命令語格納レジスタ309に格納する。そして一致比較装置302は検索命令語レジスタ301と、読出し命令語格納レジスタ309の内容を比較し、もし不一致であれば、命令語読出し制御装置303に対して、次のメモリアドレスの内容を取出すよう指示する。そこで、命令語読出し制御装置303は、アドレス更新信号発生器306を介して、読出しポインタ305の内容、つまりポインタアドレスを1アドレス分だけ加算し、次のメモリアドレス内容取出し、前述と同じ手順

(10)

によつて、検索命令を照していくわけである。そして、検索命令レジスタ301と、読出し命令格納レジスタ309の内容が、一致比較装置302によつて一致していると確認されたとき、シーケンスブロック切出し装置50に対して、出力命令一致信号307が発せられる。

次に、シーケンスブロック切出し装置50について、第4図で説明する。

まず一致命令検出装置30より出された指定回路出力命令一致アドレスデータ308は、一致アドレス記憶レジスタ500、及び、アドレスレジスタ502に、伝えられ記憶される。また、出力命令一致信号307は、シーケンスブロック先頭命令検出制御部501に伝えられる。シーケンスブロック先頭命令検出制御部501は、命令語読出し部503に起動をかける。命令語読出し部503は、アドレスレジスタ502に格納されているアドレスの内容を読出し、読出し命令レジスタ504に記憶させる。ちなみに、先頭命令検出開始時にはアドレスレジスタ502には、検索し

(11)

が1アドレス若くなる。そして次に、5022の信号で命令語読出し部503に起動がかかり、シーケンスプログラム命令語の読出しが行なわれ、読出し命令語レジスタ504に格納された後、シーケンスブロック先頭命令検出制御部501により、シーケンスブロック先頭命令語であるか否かの判定が成される。読出した命令がシーケンスブロック先頭命令でなかった場合は、以上述べた手順を繰り返し実行し、シーケンスブロック開始コード4021がはいっているシーケンス命令語402まで逆上つていく。その時、シーケンスブロック先頭命令検出制御部は、アドレスレジスタ502の内容を、先頭アドレスレジスタ506に格納する。この様な手順により、検索対象回路のシーケンスブロックの先頭アドレスが分る。

次に、シーケンスブロックの終了アドレスを検出しなければならない。この動作手順は、先頭アドレス検出とよく似ている。ただ命令語読出し開始アドレスが、先ほど同様一致アドレス記憶レジスタ500の内容を再度、アドレスレジスタ

(13)

ようとするシーケンス回路の出力命令401が格納されているアドレスデータが、はいっており、読出し命令レジスタ504には、検索の際、指定入力された出力命令401の命令語が書きこまれることになる。

さて、ここで命令語読出し部503は、命令語の読出しが終了したということで、シーケンスブロック先頭命令検出制御部501に、読出し命令レジスタ504の内容がシーケンスブロックの先頭であるか否かの判定起動信号5031を入れる。

ところで、シーケンスプログラムの先頭命令語402、403には、1シーケンスブロックの先頭命令であることを示す特殊コード4021、4031といったものがつけられており、これを検出することによつて、シーケンスブロックの開始であるか否かをチェックできるようになっている。ここで、開始命令でないと判定されると、アドレス更新信号発生器505に、アドレス戻り更新指示信号5031が出され、これによつてアドレスレジスタ502に格納されているアドレス値

(12)

502に格納することと、シーケンスブロック先頭命令検出制御部501が、読出した命令語(504の内容)は、先頭命令でないと判断したとき、シーケンスブロック先頭命令検出制御部501は、アドレス更新信号5011により、アドレス更新信号発生器505に、+1づつアドレスレジスタ502の内容を更新するよう指示する。そして最終的には、現在検索中のシーケンスブロックの、次のシーケンスブロックの先頭アドレスがシーケンスブロック先頭命令検出制御部501に検知され、その結果、アドレスレジスタ502の内容が、次シーケンスブロック先頭アドレスレジスタ507に格納される。そこで、シーケンスブロック先頭命令検出制御部501より、シーケンスブロック切出し制御部509に、シーケンスブロック切出し要求信号5012が出され、シーケンスブロック切出し制御部509は、既開発プログラム格納メモリ40より、先頭アドレスレジスタ506に収納されているアドレスから、次シーケンスブロック先頭アドレスレジスタ507に

(14)

収納されているアドレスの1番地手前迄のメモリ内容を、シーケンスブロックバッファ51に転送し同時に、転送終了信号510を、回路図変換装置65に発する。

次に、回路図変換装置65について第5図で説明する。

転送終了信号510を受けた回路図変換装置65は、シーケンスブロックバッファ51より、1語づつ、機械命令語を取出し、回路図イメージに変換後、表示装置80に表示していく。そして、シーケンスブロックバッファ51の命令をすべて処理し終えたとき、表示装置80には、1シーケンスブロックのプログラムが、ラダー回路図になって表示されている。ここでオペレータは、この回路図を見て、目的とする回路か否かを確認することができる。もし、目的とする回路でなかった場合は、再度、指定するシーケンスブロックの切出しから行なえばよい。

さて、表示されている回路は、標準回路となるべきものであり、今後再利用することが目的であ

(15)

る。その名称全体を変更する場合について、第7図で説明する。まず、オペレータは、変更モード切換えキー603を押下することにより、変更モードを、ナンバーのみの変更モードから、名称全体の変更モードに切換えることができる。この場合の入力操作は、リレー機能区分とナンバーを入力することで、制御要素の名称全体を変更していく。

同様に、制御要素の名称全体、及び、リレーシンボルの変更を行なう場合を示したのが、第8図である。操作手順は、前述の制御要素の名称変更と同様で、変更モードの切換え、カーソル位置の移動、変更後のリレーシンボル及び制御要素名称の入力、というぐあいにやっていく。第9図は、変更の処理フローを示している。

さて、以上の操作をくり返すことにより、表示装置に表示されているシーケンス回路の修正変更が行なわれていく。もし、シーケンスロジックの変更まで行なわれる場合でも、何ら問題はなく、この場合は、通常のプログラム正となるだけである。このようにして、正変更が終了するとオ

(17)

る。そのため、シーケンスロジックは、通常、変更されない。しかし、入出力点のナンバー等の制御要素の名称は、変更されることがある。

ここで、その制御要素の名称変更について述べてみる。

まず、制御要素のナンバーのみを変更する場合について、第6図で説明する。入力装置60の、カーソル逆送りキー601を押下することにより、カーソル800が、シーケンスブロックの最終出力コイル位置805より804→803→802→と、押下している間に移動していく。押下を止めればその位置でとまる。カーソル位置のリレーシンボルは、制御要素の名称だけが、変更モードになり、オペレータは、ナンバーのみを入力するだけで、変更が可能となる。変更しない場合はカーソル順送りキー602を1回のみ押下すればよい。このことにより、少ない入力操作で、かつ、誤ってシーケンスロジックを変えることなく、回路の変更が可能となる。

次に、制御要素のナンバーだけでなく、制御要

(16)

オペレータは、入力装置60中の終了キー604を押下する。すると、第5図の回路図変換装置65は、シーケンスブロックバッファ51内のプログラム命令語数つまり、表示中の回路の命令語数を算出し、サイズデータレジスタ66にセットする。同時に、シーケンスブロック登録装置70に起動信号651を発する。

次に、シーケンスブロック登録装置70について、第10図で説明する。

まず、回路図変換装置65より発せられた起動信号651は、シーケンスブロック登録装置70内の、END命令検出装置73に受けとられる。ちなみに、END命令とは、シーケンスプログラムの最後に付加されている命令語である。さて起動信号651を受けたEND命令検出装置73は、再利用プログラム格納メモリ90の先頭アドレスから、順次、END命令を検索していくわけだが、その手順は、まず、再利用プログラム格納メモリ90の先頭アドレスを記憶している先頭アドレスレジスタ731の内容を、読出しポインタ75に

(18)

セットする。そして、再利用プログラム 納メモリ 90 から、読出しポインタが示すアドレスの内容を取り出し、読出し命令語格納レジスタ 732 に納し、その内容が、END 命令か否かの判断を行なう。もし異なっていれば、アドレス更新信号発生器 74 に、信号を発する。その信号を受けたアドレス更新信号発生器 74 は読出しポインタ 75 の値を +1 更新する。そして再度、END 命令検査装置 73 は、再利用プログラム格納メモリ 90 より、読出しポインタの示すアドレスの内容を取り出し、前述と同じ手順によつて、END 命令を探していく。そして、最終的に END 命令が読出されると、シーケンスブロック格納装置 76 に対し、起動信号 733 を発する。

シーケンスブロック格納装置 76 は、その信号を受けると、シーケンスブロックバッファ 51 から、1 語づつ、命令を取込み、再利用プログラム格納メモリに格納していく。その格納の方法としては 3 つの方法がある。

第 1 の方法は、第 11 図に示す通り、再利用プ

(19)

以上の 3 つの方法があるが、ここでは、第 1 の方法について、その手順を、第 10 図で説明する。

シーケンスブロック格納装置 76 は、起動信号 733 を受けると、まず、読出しポインタ 75 の示すアドレスより、901 の END 命令を取込み今回格納するシーケンスブロックの語数分移動させ、902 の位置に格納する。移動語数は、サイズデータレジスタ 66 から取込む。それと同時に、格納アドレスポインタ 78 に、読出しポインタ 75 の内容、つまり 901 のアドレスをセットし、シーケンスブロックバッファ 51 より、命令語を 1 語取込んで、格納アドレスポインタ 78 の示すアドレスに格納する。次に、格納アドレス更新信号発生器 77 に、起動信号を出し、格納アドレスポインタ 78 の値を、+1 更新させる。そして、再度シーケンスブロックバッファより次の命令語を 1 語取込み、前述と同じ手順を行なう。このような手順を、サイズデータレジスタ 66 のデータ回数分くり返すことによつて、シーケンスブロックバッファ 51 の全命令を、再利用プログラム格

(21)

プログラムエリア 90 内に既に格納されたプログラム 900 の、最後にある END 命令 901 を、今回格納するシーケンスブロック 511 の語数分だけ、後に移動させ、902 の位置にセットしたあと、今回格納するシーケンスブロックのプログラム 51 を、901 の位置から順次 1 語づつ格納していく。901 の END 命令の移動は、今回格納するプログラムを 1 語格納するたびに、順次 1 語づつ移動してもかまわない。

第 2 の方法は、第 12 図に示す通り、今回格納するプログラム 511 に、END 命令を付加した後 903 の位置から、格納していく。

第 3 の方法は、あらかじめ、再利用プログラム格納メモリ 90 を、END 命令で、うめつくしておき、最初の END 命令の位置から、今回格納するプログラムを格納していく方法である。第 13 図は、既に格納されたプログラム 900 がある場合について、示している。つまり、最初の END 命令の位置 904 から、今回格納するプログラム 511 を格納していくのである。

(20)

納メモリに格納することができる。

以上、指定回路の切出し、表示、変更、登録という一連の処理を行うことにより、再利用プログラム格納メモリ 90 には、複数の再利用可能なシーケンスブロックからなるシーケンスプログラムができあがる。これを、ここでは標準ライブラリと呼ぶ。この標準ライブラリを、種々の既開発プログラムから作出し、複数の標準ライブラリを用意すれば、必要に応じ、結合、修正等をほどこし、目的の制御対象に合ったプログラムを作ることによつて、大幅な設計効率向上が計られる。標準ライブラリの形体は、メモリ、つまり、記憶装置であり、ICメモリ、パルプメモリ、磁気ディスク、フロッピーディスクなど、種々の記憶装置の形体となつている。

次に、その複数の標準ライブラリを結合、及び修正する方法について説明する。ここで明らかにしなければならないのは、標準ライブラリの結合と修正は、順としては自由である。つまり、複数の標準ライブラリを結合→修正の順で行い、目

(22)

的のPCsに格納してもいいし、逆に、各々の標準ライブラリを修正し、そのあと結合して、目的のPCsに格納してもよい。要は、ユーザーが一番手間のかからない手順で、目的のPCsのプログラムを作りあげればよいのである。ここでは、結合→修正の順で説明する。

まず結合であるが、標準ライブラリは、第11図～第13図で説明したように、格納されているプログラムの最後に、1個もしくは複数個のEND命令を付加した状態で保存されている。これらの標準ライブラリを結合し、一つのプログラムにまとめる場合、不必要なEND命令を削除することが必要となる。

その方法としては、次の2通りが考えられる。

第1の方法は、第14図で示すように、 n から m の標準ライブラリを単純に結合し、その後、一括して不要となった途中のEND命令を削除く。

第2の方法は、第15図で示すように、結合しようとする n から m の標準ライブラリのプログラムだけ（END命令を除いたもの）を抜き出し、

(23)

検索中の標準ライブラリから次の命令を読出し、読出し命令格納レジスタ1002にセットするとともに、格納制御部1003に起動信号1004を発する。信号を受けた格納制御部1003は、アドレス更新信号発生器1007を介し格納ポインタ1006の値を更新し、そのアドレスに、読出し命令格納レジスタの内容をセットする。以下、同様の手順にて、格納が進んでいくが、そのうち、ライブラリ検索装置1001は、検索中の標準ライブラリから、END命令を読出すことになる。そのとき、ライブラリ検索装置1001は、格納制御部1003に起動信号を入れず、次の標準ライブラリの検索、及び、命令語の読出しに移る。その結果、END命令を除いて、次の標準ライブラリの格納に移ったことになる。以上の手順を指定された標準ライブラリの数だけくり返し行ない、最後に指定された標準ライブラリのEND命令が、読出されたときには、ライブラリ検索装置1001は、格納制御部1003に、起動信号を短し、END命令を格納させる。このような方法で、結合プロ

(25)

それぞれ結合したあと、最後にEND命令を付加する方法である。例えば、第2の方法について第16図で説明すると、結合しようとする標準ライブラリの n を、入力装置110から入力する（当然入力データは、 n とは限らず標準ライブラリ名称でも、何でもよい。要は、目的となる標準ライブラリを指定できればいい。）。すると、ライブラリ検索装置1001は、指定された n 順に、標準ライブラリを検索し、命令語を1語読出して、読出し命令格納レジスタ1002にセットする。同時に、格納制御部1003に起動信号1004を入れる。格納制御部1003は、結合プログラムエリア130の先頭アドレスを記憶している先頭アドレス記憶レジスタ1005の値を格納ポインタ1006にセットし、読出し命令格納レジスタ1002の内容を、格納ポインタ1006の示すアドレス、すなわち、結合プログラムエリア130の先頭に格納する。そして、ライブラリ検索装置1001に、格納終了信号1008を発する。信号を受けたライブラリ検索装置1001は、

(24)

グラムエリア130には、第15図で示すような、複数の標準ライブラリが結合されたものができあがる。この結合プログラムを新たな標準ライブラリとすることも可能であるし、このまま、目的のPCsに格納し、そのまま利用することも可能である。しかしここでは、更にその結合プログラムに修正を加える場合について説明する。

そもそも、この結合プログラムは、標準プログラムとして再利用可能なシーケンスブロックの集合体であるから、シーケンスのロジック変更によって及ぶ修正は、通常考えられない。そのような大きな変更については、一般のプログラム修正と同じことであり、本発明の主旨ではない。ところがまったく修正がないとは言いきれず、例えば、第5図で説明した回路図変換装置による入出力のナンバーなどの、制御要素の名称変更などは、考えられる。そこでここでは、制御要素の名称変更を一括して行なう場合について説明する。

制御要素の一括変更は、第5図の回路図変換装置によるシーケンスブロックの修正（第6～8図）

(26)

で説明したように、①制御要素のナンバーのみの変更、②制御要素の名称全体（リレー機能区分とナンバー）の変更、③制御要素の名称と、リレーシンボルを含めた変更、の①～③が考えられるがここでは②の 合を例にとる。

第17図は、一括変更装置を示したものである。入力装置160より、変更前と変更後の名称を、第18図の表示装置170に示すように入力し、**START** キーを押下すると、第17図の命令語変換装置1501は、変更前と変更後の機械命令語を作成し、変更前命令格納レジスタ群1502と、変更後命令格納レジスタ群1503にセットする。そして、一括変更制御部1504に対し、起動信号1505を送る。起動信号を受けた一括変更制御部1504は、結合プログラムエリア130の先頭アドレスを記憶している先頭アドレス記憶レジスタ1506の値を、読出しポインタ1507にセットし、結合プログラムエリア130より、命令語を読出し、読出し命令格納レジスタ1509に格納する。そして読出した命令語が、変更前命

(27)

を作成。更に、その標準ライブラリの結合、修正を行なうことにより、目的となるプログラムが作りあがる一連の説明をしてきた。

本発明が特に有効となる例をあげてみると、まず、機械セツトメーカーの工作機械制御プログラムがあげられる。工作機械の制御プログラムは、一般に、80%が共通、もしくは、制御要素の名称変更の修正を加えてリピータ使用可能であり残りの20%が、ユーザー要求による新規作成となるので、機械セツトメーカーの設計者は、本発明を使用すると、80%のプログラムを標準ライブラリとして保存し、再利用できることになる。80%のプログラムの作成作業が、削減できるのである。その他の例としては、ダム制御や、上下水制御のポンプ制御にも有効である。ダムや上下水の制御には、複数台のポンプが使用され、入出力の割付けが異なるだけで、制御そのものは、まったく同じものである（第19図参照。ここでは、制御要素名称は異なるが、制御ロジックは同一である例を示している。）。このような場合は、機

(29)

命令格納レジスタ群、1502の内容と比較していく。もし、一致するものが無い場合、アドレス変更信号発生器1508を介して、読出しポインタ1507の値を更新し、その更新されたアドレスから再度、命令語を読出す。そして、前回と同様に、変更前命令格納レジスタ群1502の内容と比較し、もし一致するものがあれば、変更後命令格納レジスタ群1503内の、該当する変更後命令を読出しポインタ1507の示すアドレスに格納する。このような処理をくり返し行ない、最終的に、読出した命令がEND命令であつたとき、一括変更の処理が終了する。

以上のようにして、標準ライブラリの結合、及び、修正をほどこされた結合プログラムは、プログラム作成装置により、目的となるPCsに格納される。格納については、通常行なわれるプログラム作成装置からのプログラムローディングと同じである。

以上、既開発プログラムから、再利用可能なものを抽出し、修正を加えたのち、標準ライブラリ

(28)

準ライブラリを作成して、制御要素名称の一括変更をほどこせば、そのまま、再利用可能となる。

ほかにいろいろな例が考えられるが、要は、再活用可能なプログラムが多い制御であればあるほど、その再活用可能なプログラムの量に比例して、本発明の効果がでてくるのであり、さらに言えることは、PCs 制御システムは、再活用可能なプログラムの多い制御システムとなることが、しばしばなのである。

従来は、標準プログラムが、書類ベースで管理保存されており、それを利用するときは、その図面を参照しながら、プログラム作成装置で、再度プログラムを打込む必要があつた。しかし本発明を使用すれば、そのプログラムを打込む作業が無くなり、大幅な設計生産効率向上がなされる。たとえば、あるPCのプログラムで、90%が標準プログラム、残り10%が新規作成プログラムとなつていた場合、本発明を使用することにより従来比90%の、プログラム作成作業低減が成されるわけである。

(30)

【発明の効果】

本発明によれば、標準プログラムを新規に作成することなく再利用できるので、大幅なプログラミング作業能率の向上を図ることができる。

4. 図面の簡単な説明

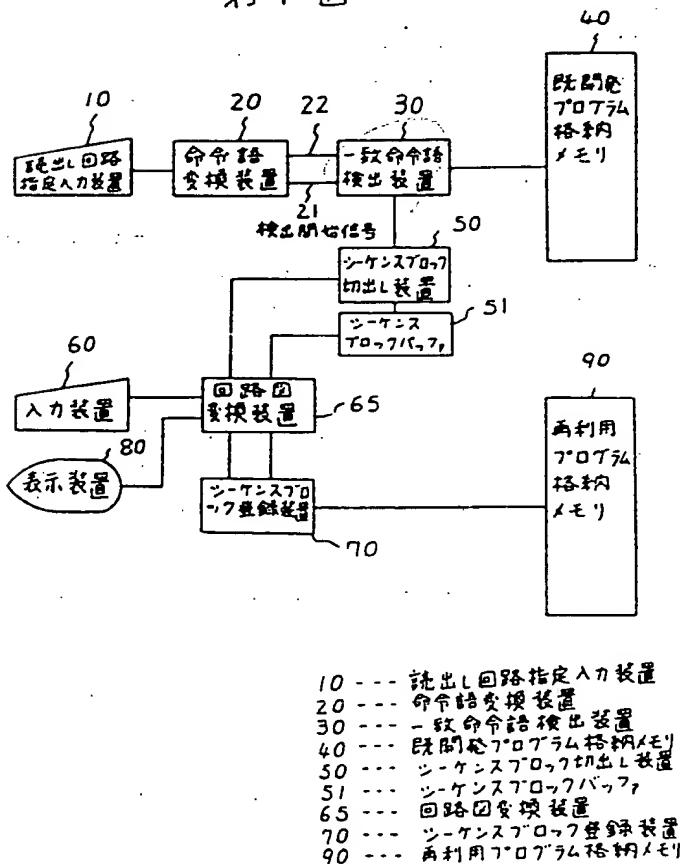
第1図は本発明におけるプログラム作成装置の全体構成図、第2図はシーケンスブロック、第3図は一致命令語検出装置、第4図はシーケンスブロック切出し装置、第5図は回路図変換装置、第6～第9図は回路図変換装置の回路修正を示す図、第10～第13図はシーケンスブロック登録装置とその登録方法について示す図、第14～第16図は標準プログラムライブラリの結合方法と結合装置を示す図、第17、18図は一括変更装置、第19図はポンプ制御プラントの例について示す図である。

10…読出し回路指定入力装置、20…命令語変換装置、22…一致命令語検出装置、30…既読プログラム格納メモリ、40…既読プログラム格納メモリ、50…シーケンスブロック切出し装置、51…シーケンスブロックバッファ、60…入力装置、65…回路図変換装置、70…シーケンスブロック登録装置、80…表示装置、90…再利用プログラム格納メモリ

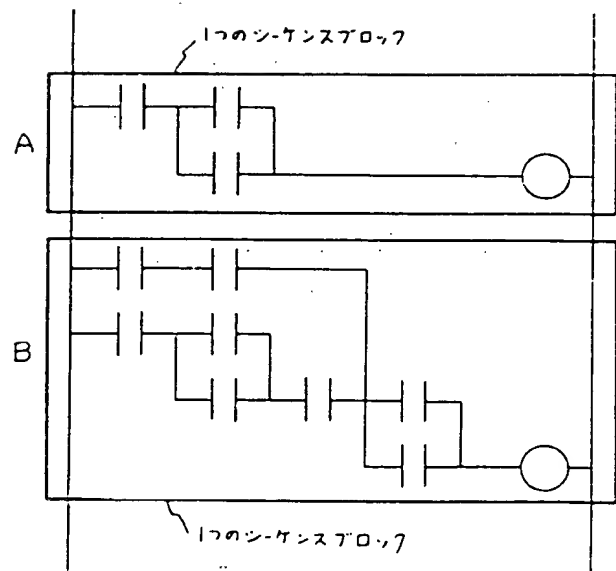
(31)

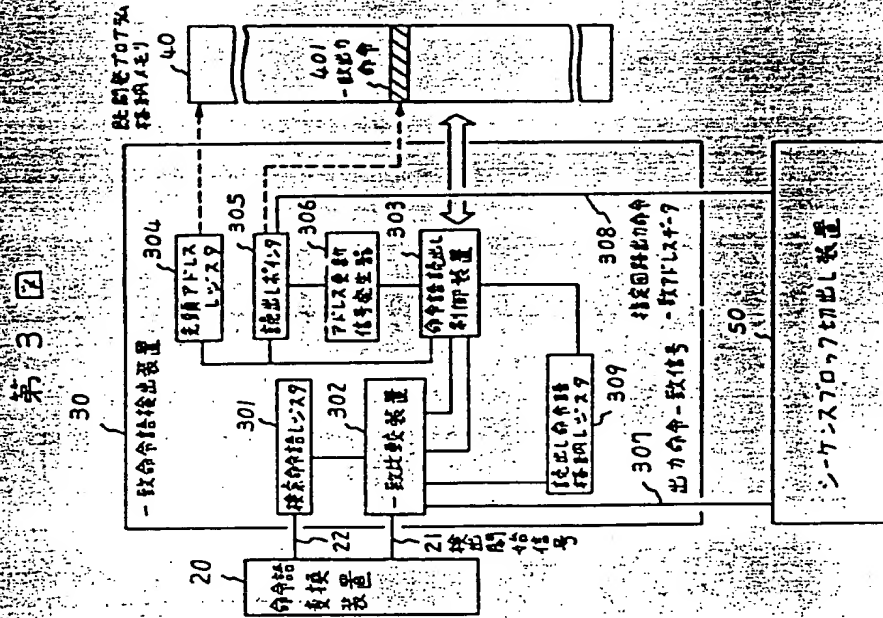
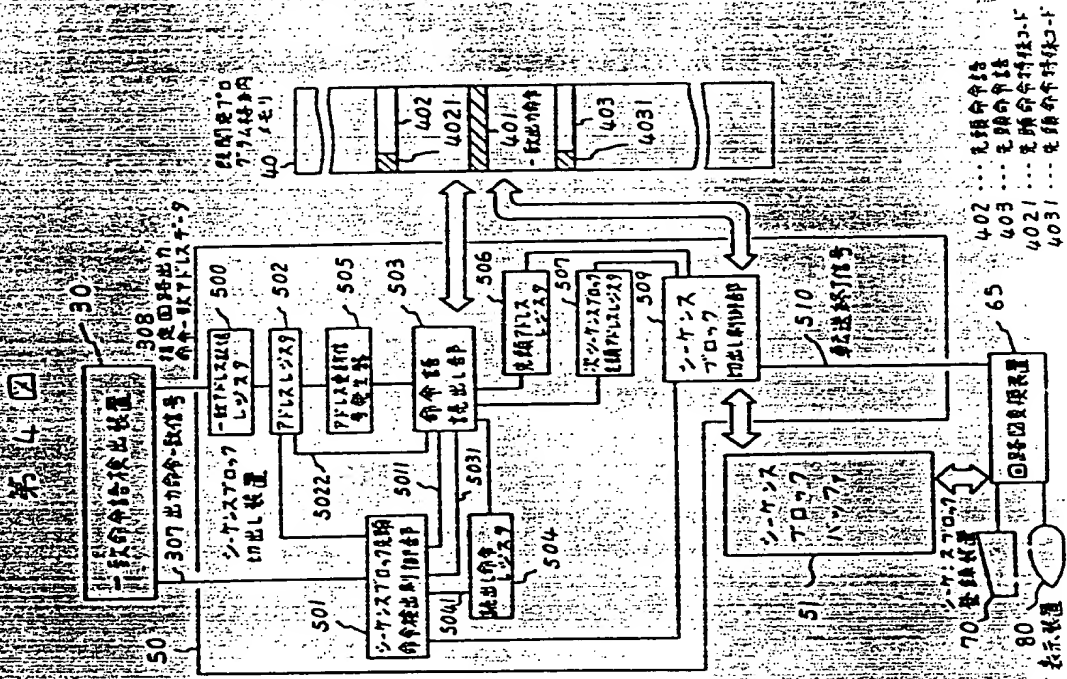
(32)

第1図

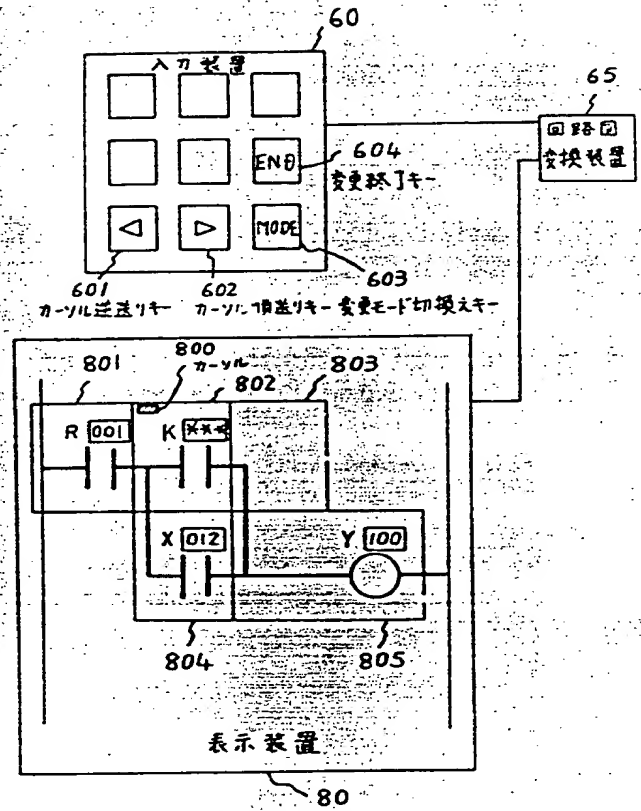


第2図

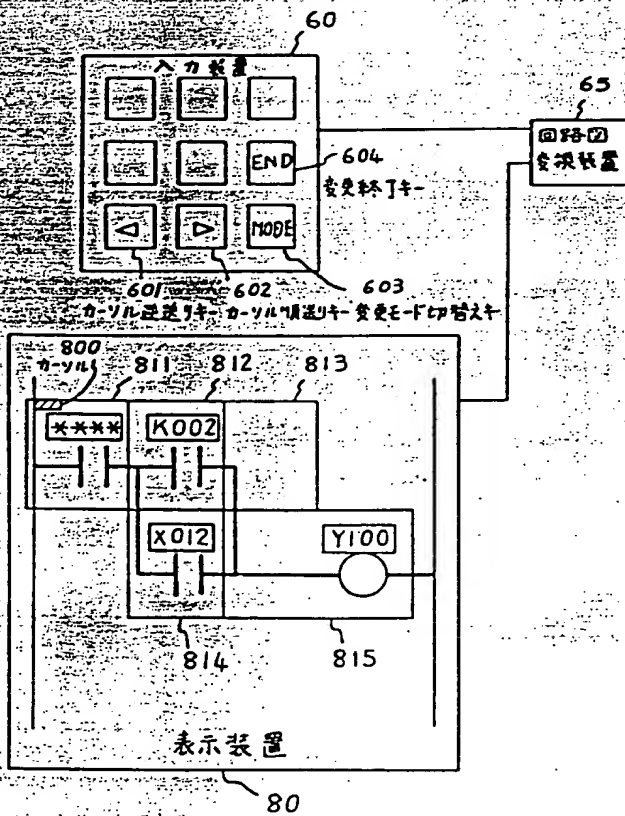




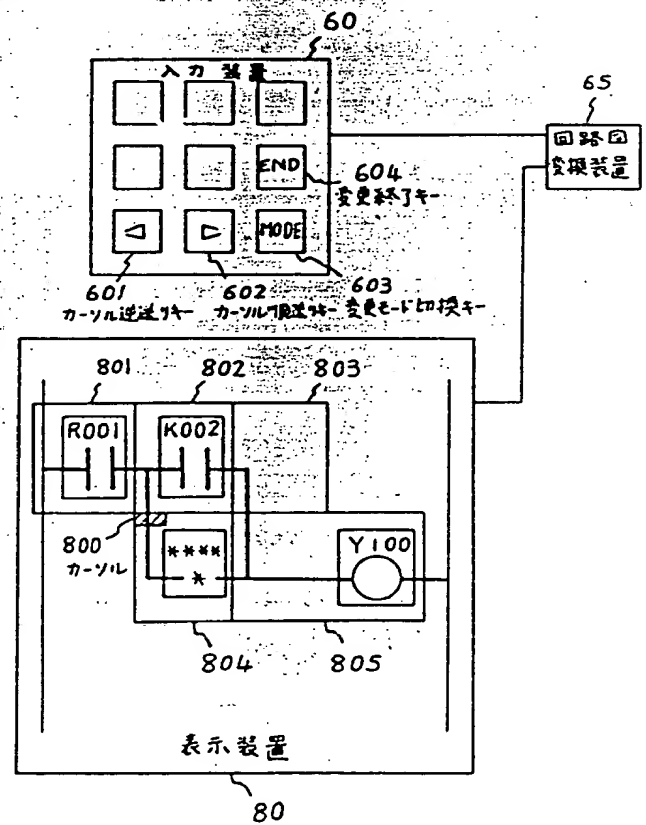
第6図



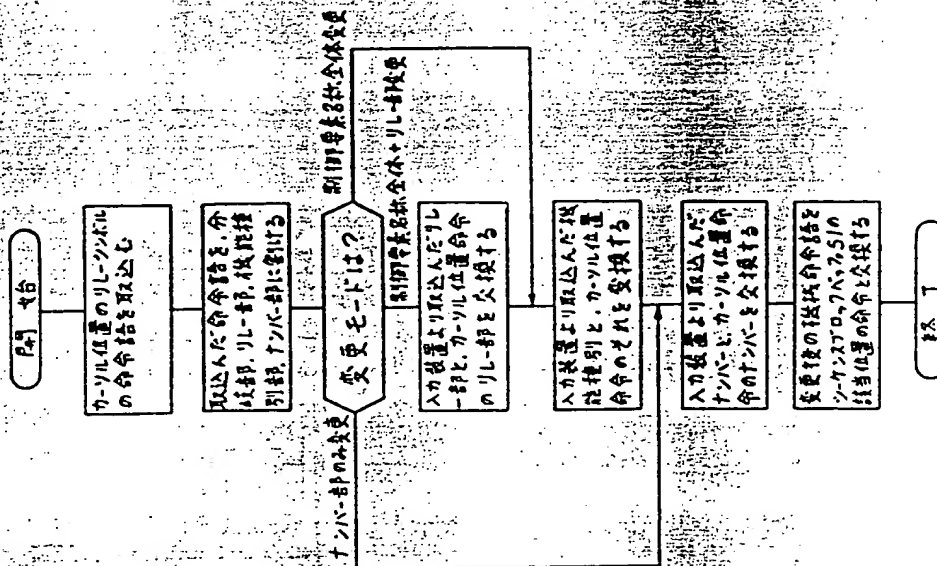
第7図



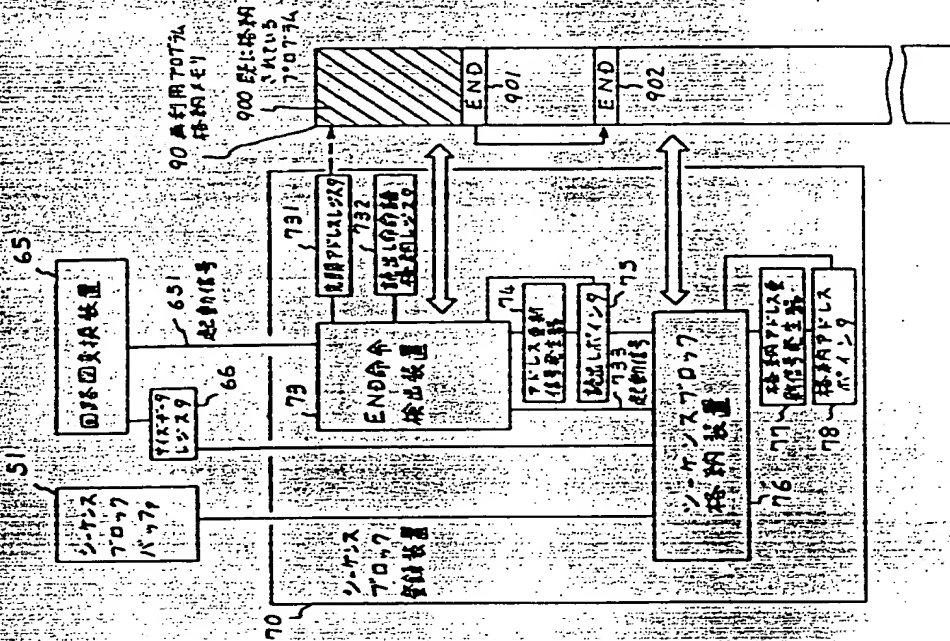
第8図



第 9 圖

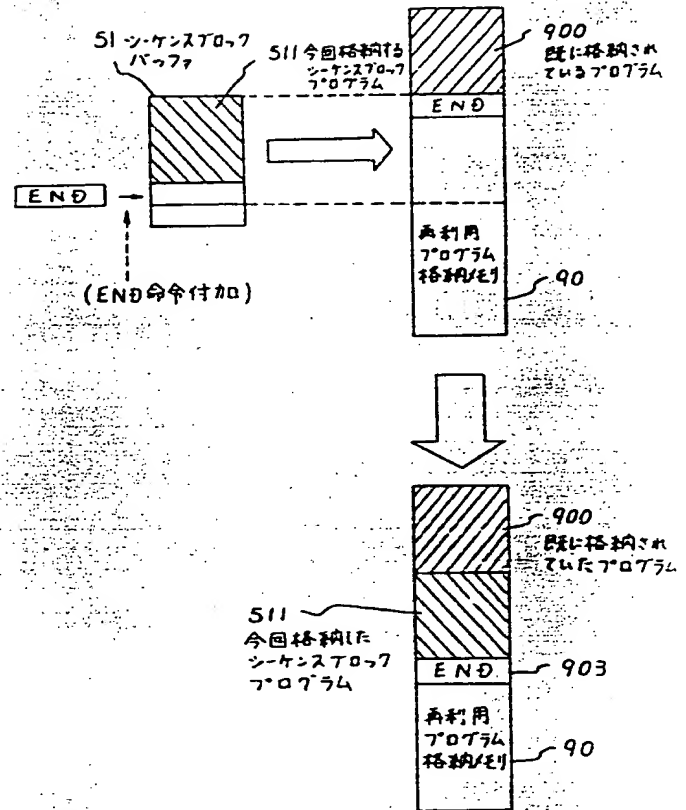
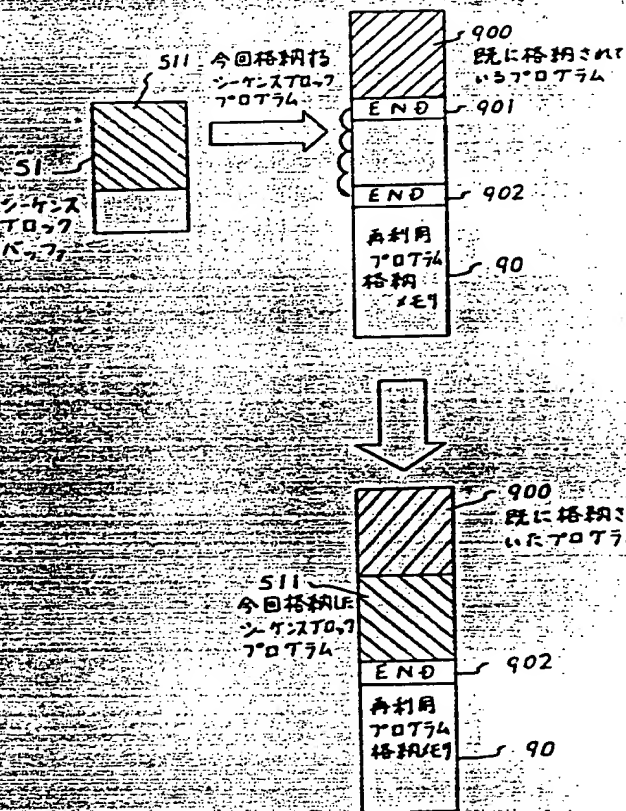


日一十

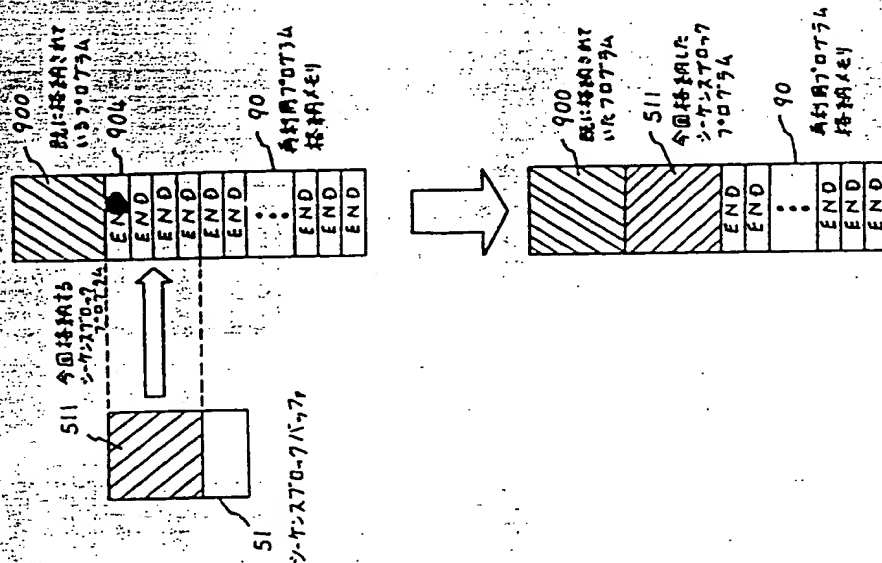


第11図

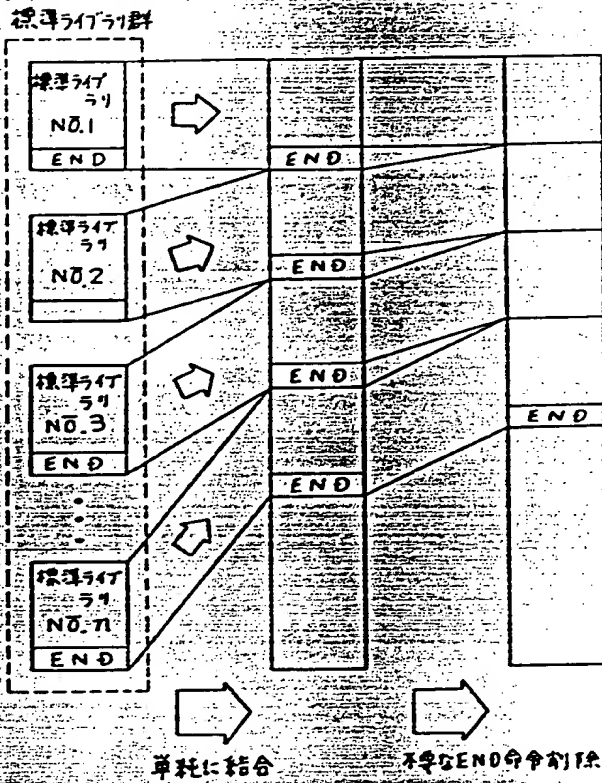
第12図



第13図



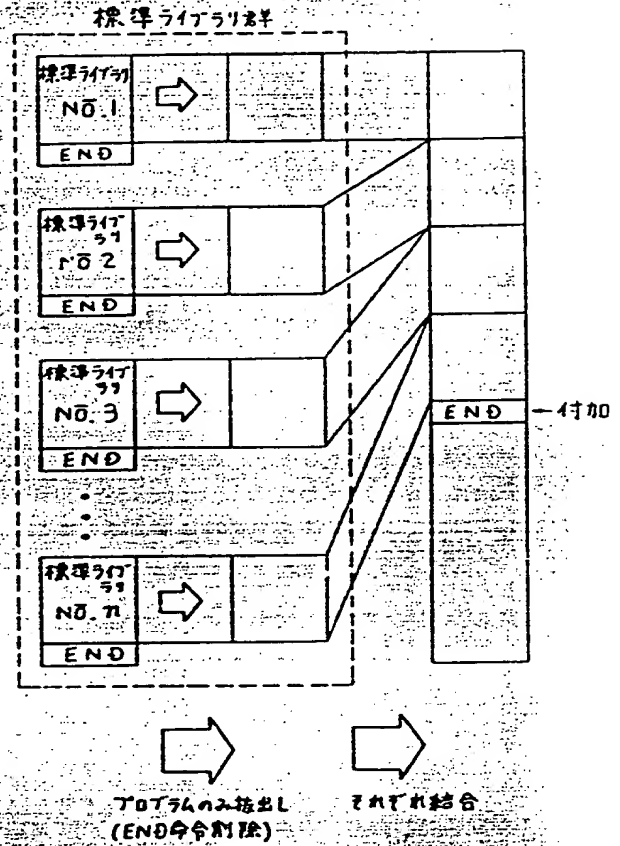
第14圖



單純結合

不要在END命令删除

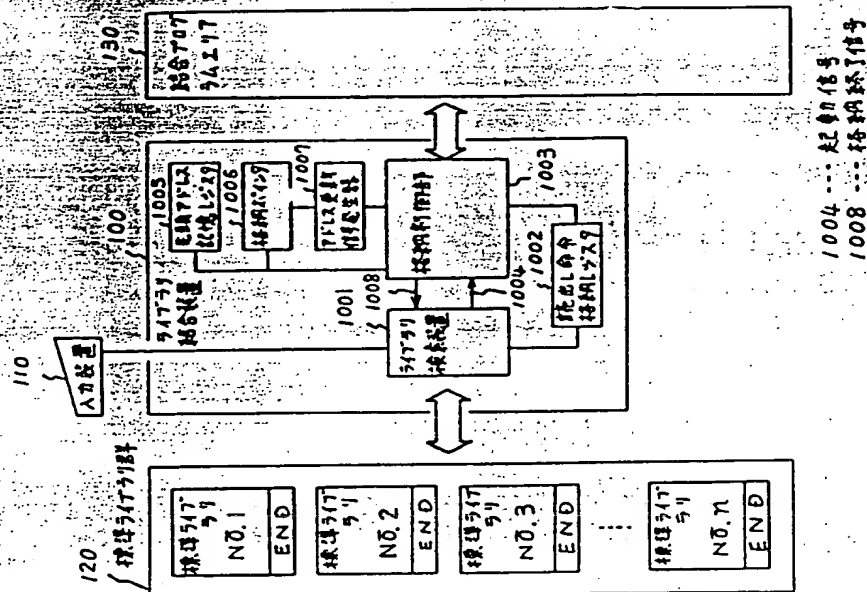
第 15 回



TOPランのみ抽出
(END命令削除)

それそれ結合

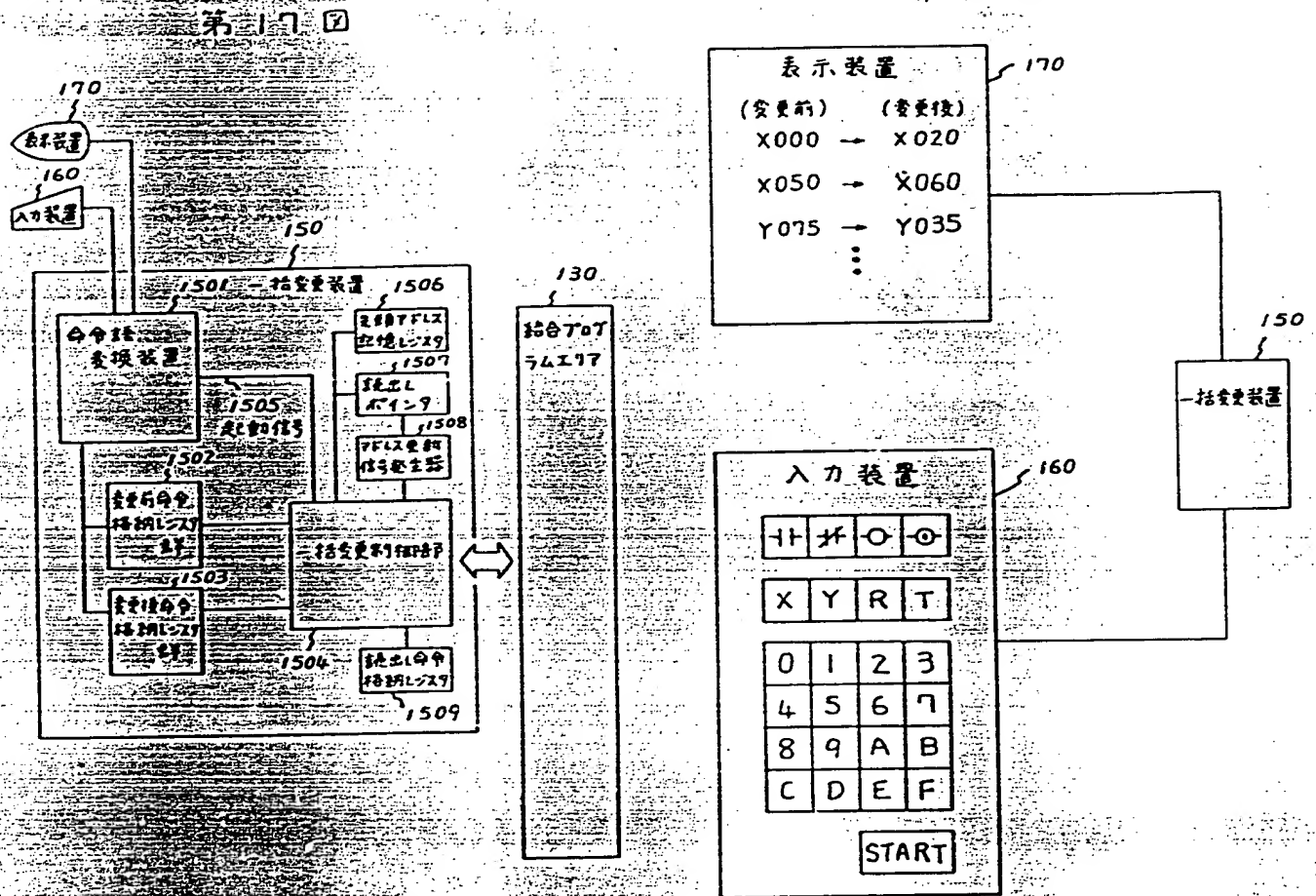
第 16 圖



1004 -- 起重信号
1008 -- 格纳铃了信号

1008 -- 格

第 18 回



第四

